Assistance to Agent-Based µ-Tools Development for a Co-operative Design Platform

Alain-Jérôme Fougères

Laboratory M3M, University of Technology of Belfort-Montbéliard (UTBM)
rue du château Sévenans – 90010 BELFORT - France
E-mail : alain-jerome.fougeres@utbm.fr

Abstract: We report in this paper the assistance design of applications to actors of design process whom we call micro-tools (µ-tools). The µ-tools are well adapted to instrument the co-operative activities integrated in virtual spaces of the design. Those can be integrated on a software-agent platform (*PLACID*) adapted to the innovating and distributed design mechanical systems. The development of these µ-tools is a collaborative activity bringing together the future users and the developers. This collaboration is not always easy. Also, we try to propose possibilities of assistance throughout the development of the µ-tools, starting from the interactions and the natural expression of the various actors.

Key words: µ-tools, assistance design, natural language processing, cooperative work.

1- Introduction

The objective of this article is to report the design of assistance applications to group of actors of software engineering whom we call micro-tools (µ-tools). Those can be integrated on a cooperative platform. The concept of µ-tools consists of software applications which are light, easy to use, insertable in a shared environment, connected between them using a database. We developped an agent-based platform (*PLACID, [F4]*) with an aim of bringing an assistance to co-design work. The principal characteristics of the agents (autonomy, adaptability, co-operation and communication [F1]) make it possible to effectively manage distributed, heterogeneous and autonomous components, and to facilitate the exchanges of information and the resource sharing between the components (interaction, communication and co-operation). The agents are of type: application (µ-tools or other tools of assistance to co-design), coordinator/mediator, system and interface.

The development of µ-tools follows a process which we named *IDI* (Identification, Design, Integration). To facilitate the interactions between the various actors implied in these developments we design tools based on techniques of *NLP* (Natural Language Processing). The design (software for example) assisted by NLP is an old research topic [L1]. In this short paper we will focus on the first phase of ICI methodology, the identification of the µ-tools.

This article will be structured as follows: in section 2 we present the various concepts implied in cooperative work, mainly the context of software co-design and the concept of µ-tool. Section 3 starts by presenting our objectives of assistance for the µ-tools development process, then we illustrate our first experiments on the phase of identification. Finally, in section 4, we discuss the prospects of our work.

2- µ-tools to instrument co-operatives acitvities

One of the major topics in the field of *CSCW* (Computer Supported Cooperative Work) [SB1] is the development of groupwares [EG1]. By definition a groupware is a software which assists a user group for the realization of joint project. The members of group collaborate either at the same moment (synchronous activity), or at different times (asynchronous activity). The applicability is numerous: products design, teaching, trade, games. It is usual to present the co-operative information systems like being able to answer to the needs of the co-operation: to facilitate the resource sharing, to assist the coordination, to improve the communication of group, to support the individual motivation, and to support the development of the organization.

2.1- The Concept of µ-Tool

The concept of µ-tool [VH1] is opposed to the current tendency tools of design, which are often heavy, prescriptive, and finally not used. These tools must be ideally (see Figure 2):

- easy to learn (a few minutes) and easy to use;
- simple (even if they are based on a complex theory);
- easily programmable and easily modifiable, (including by the users themselves) usable in an opportunist way;
- autonomous, but also reactive when they are defined for co-operative processes; we name then them *CMT* (Co-operative Micro-Tools [F6]).

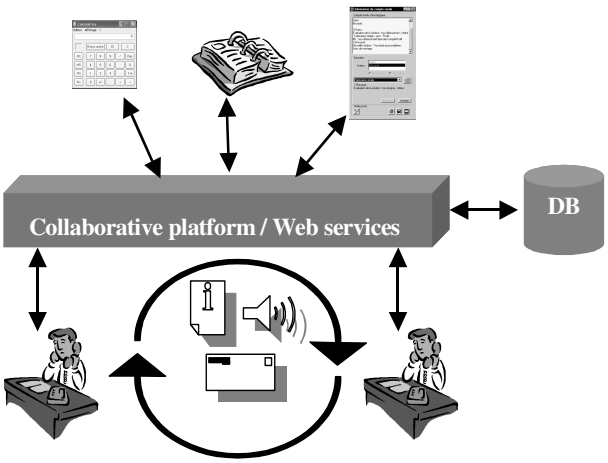


Figure 1 : The concept of µ-tool.

2.2 A Process to develop µ-tools

The µ-tools make it possible to divide a software into modules adapted to the achievement of simple tasks. Several tools of same types can fulfill more complex functions. We defined a complete process of development of µ-tools making cooperate the actors trades (Figure 2). This process begins with the activity analysis and leads on the corresponding software products and seven documents constituting the memory of their designs. Three great phases structure this process named *IDI*:

- **identification** of µ-tools, referring on the higher levels of engineering system and requiring a large collaboration of all the actors, consists in identifying among the tasks which make a specific activity, those which could be instrumented, then to specify them;

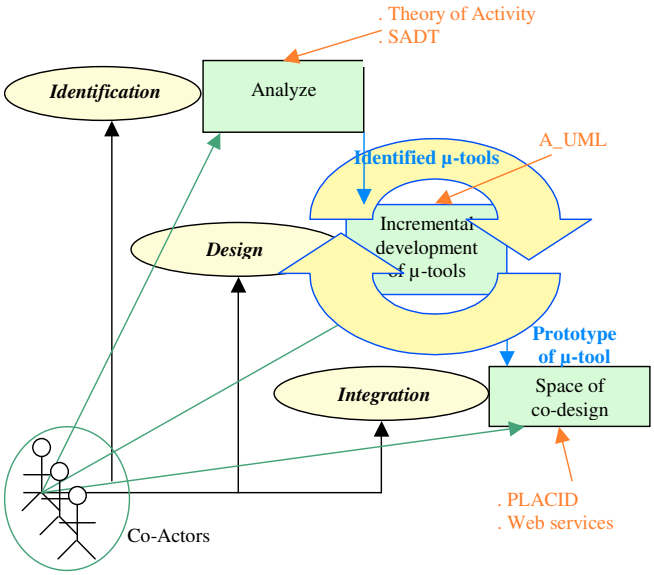


Figure 2 : Development process of a µ-tool of design

- **design** of µ-tools, according to an incremental approach which facilitates the permanent dialogue between all the actors of the process of development, to define the architecture of the µ-tool and these components, to develop them (UML/Java), to test them, and finally to validate the user interface;

- **integration** of µ-tools to *PLACID* platform in charge of exchanges management and information sharing.

3- NLP to assist the identification of µ-tools

3.1 A process for natural assistance

This research is motived by desire to develop tools which will be able to help together designers and software engineers to identify the µ-tools to be designed for the instrumentation of an innovating activity. For this move towards assistance, we propose a sequence of process from the expression of needs in natural language, to the production of semantic diagram (functional description oriented), translatable in SADT form (Figure 3).

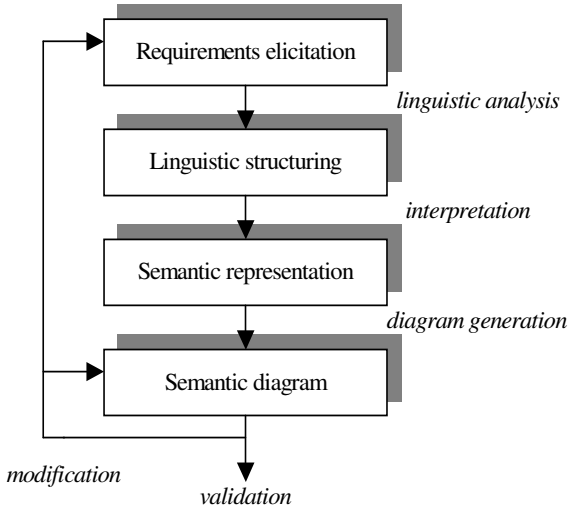


Figure 3 : Process of assisted identification of µ-tools

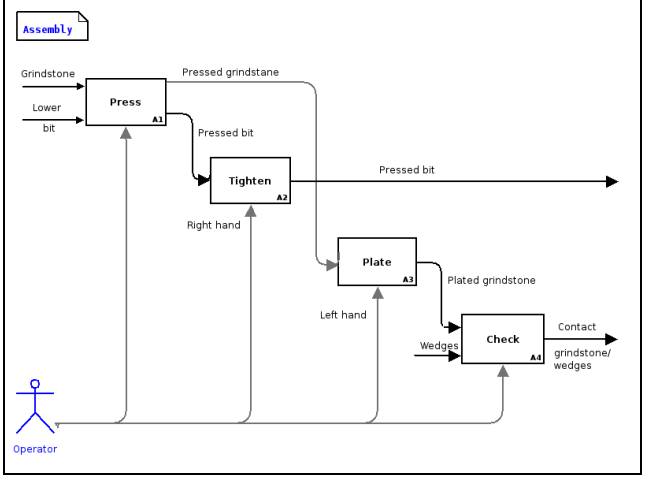
Text of the first task of requirements
Assembly: *The operator presses the grindstone on the lower bit. He tightens using the mechanical lever the bit with the right hand while it plates the grindstone with the left hand. He checks the contact between the grindstone and the wedges.*
SADT actigram for this task


Table 1 : Current result of analysis of the task "Assembly"

3.2- An illustration of assisted identification

Our experiments was based on a text of requirements elicitation written by Rabardel in [R1] (see an extract translated in english in Table 1). This text is used to illustrate an activity analysis on a machine-tool using the model of the situations of instrumented action (*SIA*). The analysis makes it possible to translate an approach into term of activity in an approach in technological terms. The various actions are gathered in tasks potentially instrumentables by a µ-tool, in an environment of simulation for example.

The process of language analysis includes 2 phases: an knowledge acquisition of the domain, and the linguistic analysis itself which we develop in the continuation of paper. This second phase is generally sub-divided into five analysis: morphological, lexical, syntactic, semantic, pragmatic. For the semantic representation we have taken inspiration from the case grammars which determine the different thematic roles taken by the components of a sentence with the help of information acquired about word-order, about prepositions, verbs and context. The analyser determines the way in which the nominal groups of a sentence are bound to the verbs: the semantic role specifying how an object participates in the description of an action [F3].

***SIA* model** *Concepts* / Sentences to be analyzed	**Subject (S)** *Agent*	**Action (A)** *Method*	**Object (O)** *Object*	**Instrument (I)** *System*	**Interaction** *Association*
"The operator presses the grindstone on the lower bit"	operator	press	grindstone lower bit		grindstone/bit
"he tightens using the mechanical lever the bit with the right hand"	operator	tighten	bit	mechanical lever right hand	bit/lever bit/right hand
"while it plates the grindstone with the left hand."	operator	plate	grindstone	left hand	left hand / grindstone
"He checks the contact between the grindstone and the wedges	operator	check	contact grindstone, wedge		grindstone/wedge

Table 2: Analysis starting from *SIA* model and associated software concepts

The linguistic treatment is object oriented and carried out in JAVA/UML (Figure 5). Each sentence of the requirements consists of a list of words belonging to a grammatical category. The grammatical rules correspond to forms of simple sentences. Semantic interpretation is produced by the rules of a cases grammar [F5]. This enables us to directly translate a sentence into terms of objects, agents and relations. Adapted to our agent approach (AUML [OP1]), the analyzer distinguishes indeed the entities corresponding to agents or objects.

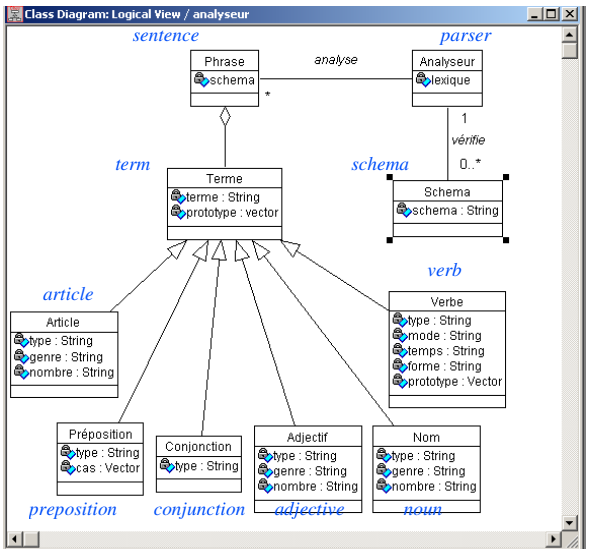


Figure 5 : Partial class diagram of syntax/semantic analysis.

The process illustrated by this example is automatic, which is made possible by the analysis of simple sentences. Of course, for a corpus of complex sentences, it is preferable to implement an approach of assistance to users. Moreover, our analyzer treats French sentences. It would be interesting to develop rules (schemas) for several languages. In the state of the prototype (and for simple sentences) that would be rather easy since the user can modify or change the lexicon and the grammatical rules.

To finish this presentation, figure 6 illustrates the whole of these processings applied to the example given in table 1. On the basis of graph result, designers and software engineers can discuss the interest to develop or not a µ-tool. Here a proposal to instrument the tasks "to press" and "to check

5- Conclusion

Starting from a study undertaken on the prospects for co-operative work and the agents technologies [F2, F3], it is arisen a whole of promising concepts which were used for the definition of the PLACID platform. Following this study we specified then developed several sets of µ-tools to assist the collaborative design process [WF1].

Now we work on the design of assistance tools (even µ-tools) of development of µ-tools. These assistance tools use techniques of NLP. As it is difficult to claim with a completely automatic NLP, we prefer to propose tools of assistance. The activity of development of µ-tools being well apprehended (process *IDI*), we thought of re-using the concept of µ-tool to assist it. Thus a whole of µ-tools could make it possible a collective multi-trade to identify, design and integrate a µ-tool (in particular for the edition of specifications and class diagrams in UML [F5]). This corresponds to our prospects.

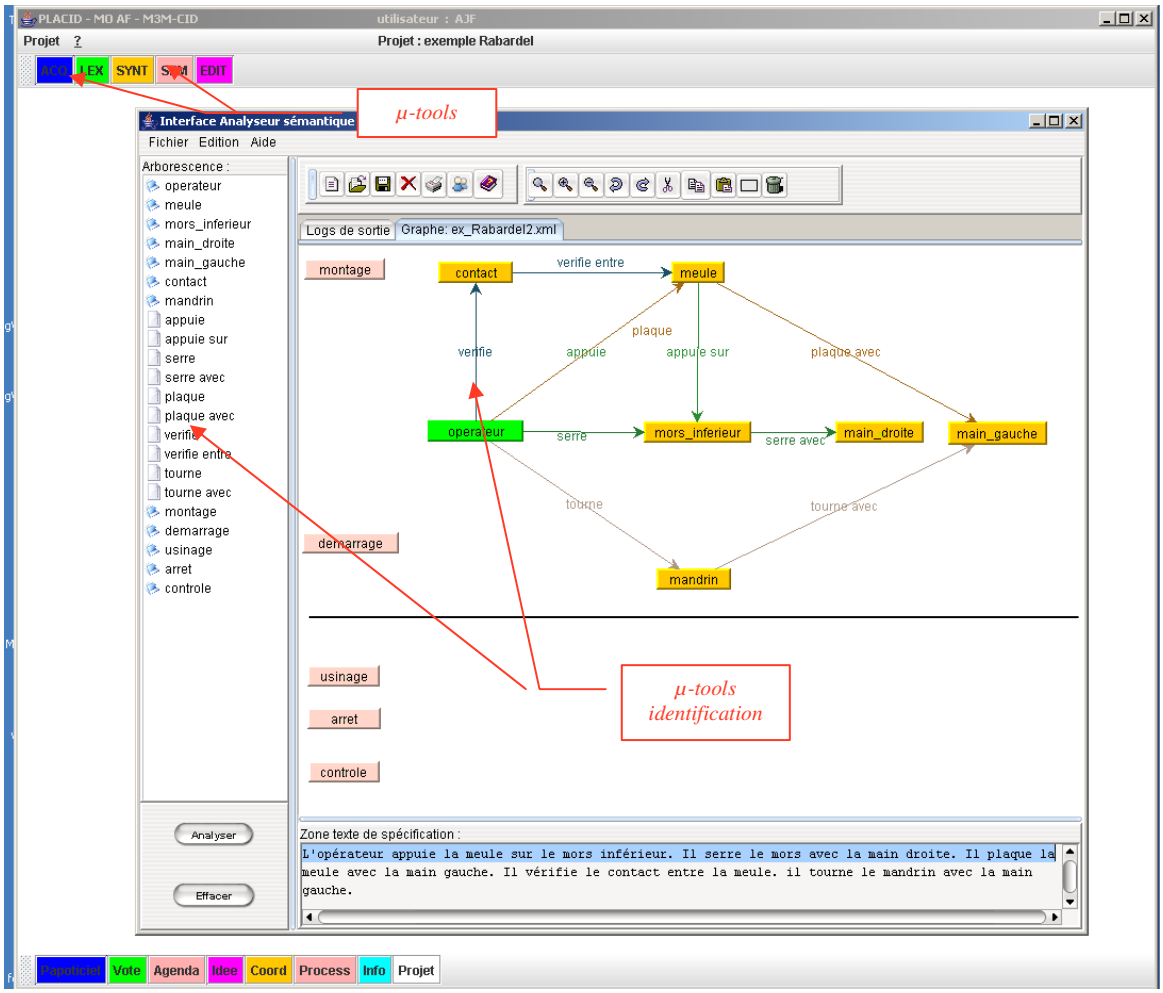


Figure 6 : µ-tools of edition of specifications diagrams

7- References

[EG1] Ellis, C.A., Gibbs, S.J. & Rein, G.L., Groupware: some issues and experiences, Com. of ACM, 34(1), p. 38-58, 1991.

[F1] Ferber, J., Multi-Agent Systems: towards a collective intelligence, Addison-Wesley, Reading, 1998.

[F2] Fougères A.-J. Formal specifications building from specifications written in natural language. In proceedings of HCP'99, Brest, France, 1999.

[F3] Fougères A.-J., Model of cognitive agents to simulate complex information systems, IEEE Int. Conf. on Systems, Man and Cybernetics, (SMC'02), Hammamet, Tunisia, October 6-9 2002.

[F4] Fougères A.-J., Agents to cooperate in distributed design, IEEE Int. Conf. on Systems, Man and Cybernetic, (SMC'04), The Hague, Netherlands, October 10-13 2004.

[F5] Fougères A.-J., Du langage naturel à la spécification - Application à la spécification de services de télécommunication, SETIT'04, Sousse, Tunisie, 15 mars 2004.

[F6] Fougères A.-J. Agent-based micro-tools development for a co-operative design platform. ITI 3rd Int. Conf. on Information and Communication Technology, Cairo, Egypt, December 5-6, 2005

[K1] Kuutti, K. Activity Theory as a Potential Framework for Human-Computer Interaction Research. In Context and Consciousness. Activity Theory and Human-Computer Interaction, MIT Press, Cambridge, p. 17-44, 1996

[L1] Loomis, T. Software design issues for natural language processing. Machine Translation, 2(4), p. 219-230, 1987.

[OP1] Odell, J., Parunak, H.V.D. & Bauer, B., Extending UML for agents, Proc. of the AOIS Workshop at the 17th Nat. conf. on Artificial Intelligence, Austin, Texas, 2000.

[R1] Rabardel P., Les hommes et les technologies. Approche cognitive des instruments contemporains. Armand Colin Editeur, Paris, 1995.

[SB1] Schmidt, K. & Bannon, L., Taking CSCW seriously, Computer Supported Cooperative Work Journal, 1(1), 1992.

[VH1] Van Handenhoven, E. & Trassaert, P., Design knowledge and design skills, International Conference on Engineering Design (ICED 99), Munich, 1999.

[WF1] Weité, P.-A., Fougères A.-J., and Gazo C. Les micro-outils, vecteur d'appropriation des nouvelles méthodologies de conception et d'innovation. In Evaluation et décision dans le processus de conception, sous la direction de B. Yannou et E. Bonjour, Traité IC2, Hermès-France, p. 135-149, 2006.